Architecture/Design for IOC-1

TRANSIMS Team

Los Alamos National Laboratory

5 April 1995

Abstract

- The purpose of this presentation is to outline the architecture and high-level design for TRANSIMS.

- Although it specifies the basic capabilities for IOC-1, it also provides a framework into which to incorporate the future IOCs.

- The basic goal is to make sure that the various components of TRANSIMS are effectively integrated, both for IOC-1 and beyond, so that TRANSIMS remains flexible, expandable, portable, and maintainable throughout its lifetime.

The Importance of Architecture

- Architectural considerations become more important for larger projects:

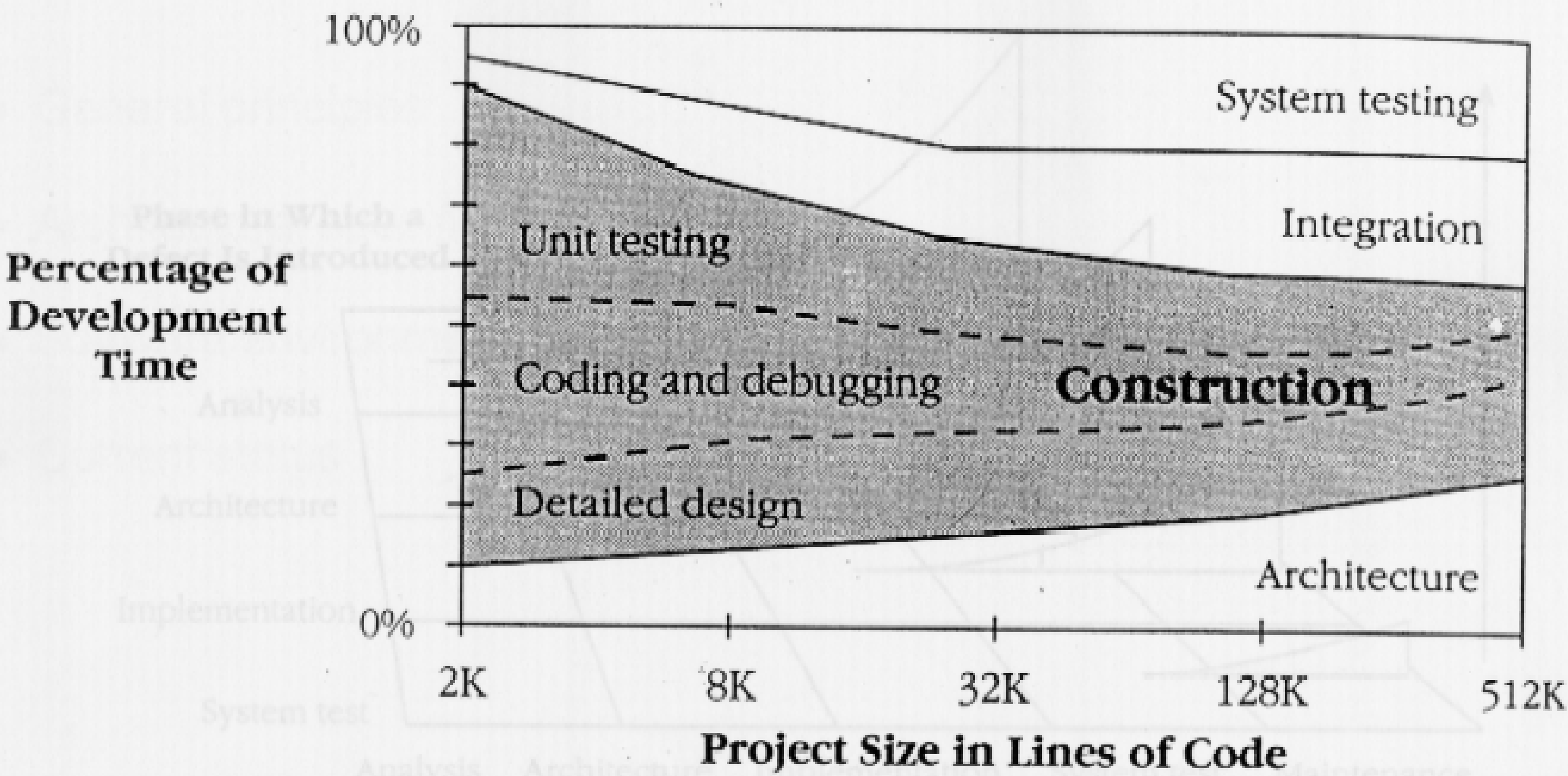
  "On a small project, construction is the most prominent activity by far, taking as much as 80 percent of the total development time. On a medium-size project, construction is still the dominant activity but its share of effort falls to about 50 percent. On very large projects, architecture, integration, and system testing each take up about as much time as construction."[*]

- It pays to do things right the first time:
  - Data from TRW shows that a change in the early stages of a project, in requirements or architecture, costs 50 to 200 times less than the same change later, in construction or maintenance.[†]
  - Researchers at IBM found that purging an error by the beginning of design, code, or unit test allows rework to be done 10 to 100 times less expensively than when it's done in the last part of the process.[‡]

---

[*] *After S. McConnell, 1993.*
[†] *After Boehm and Pappacio, 1988.*
[‡] *After Fagan, 1976.*

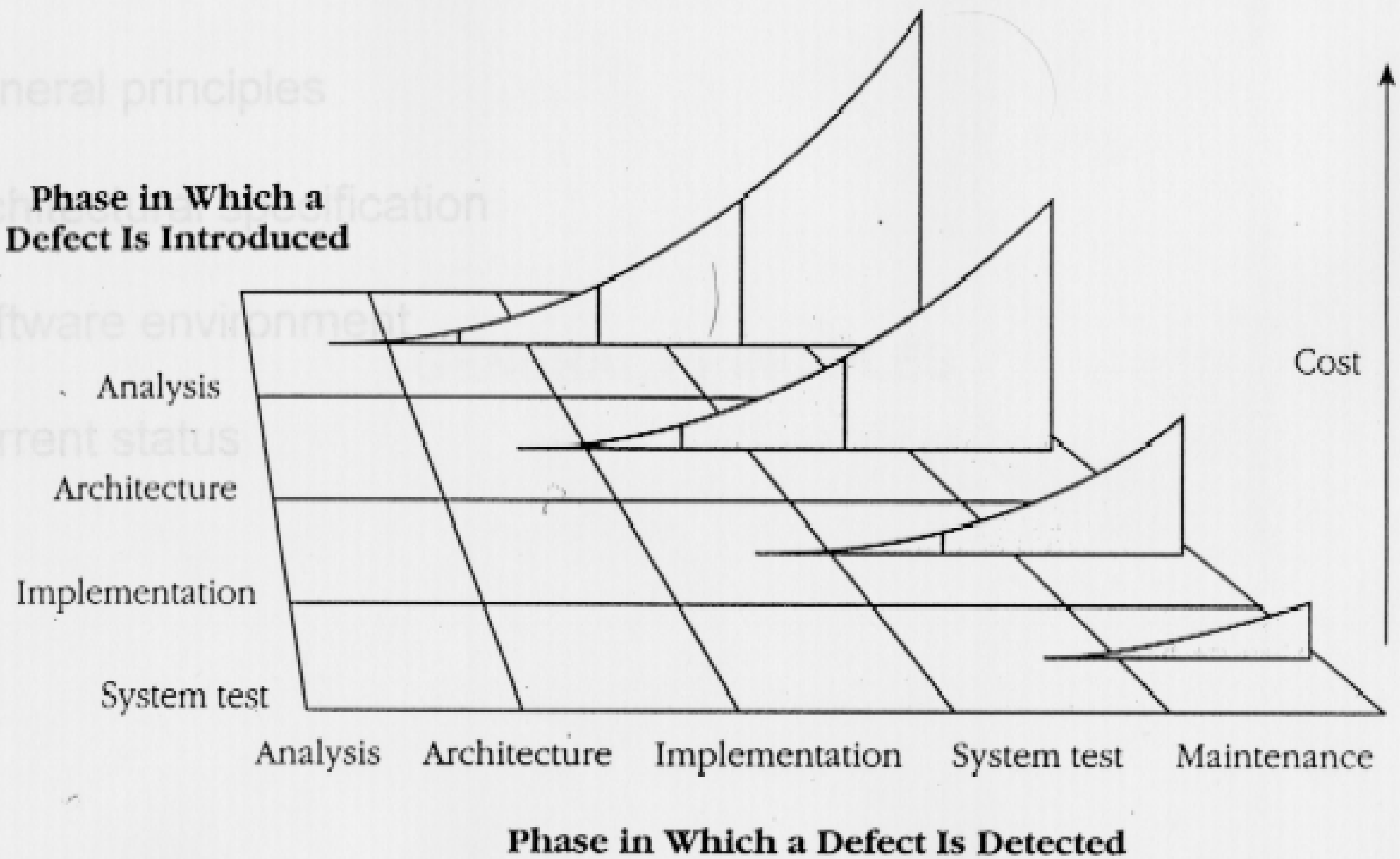
Activity Proportions and Size*



Sources: Brooks 1975; Albrecht 1979; Glass 1982; Boehm, Gray, and Seewaldt 1984; Boddie 1987; Card 1987; McGarry, Waligora, and McDermott 1989.

After S. McConnell, 1993.

Cost of Fixing Defects*



* After S. McConnell, 1993.

Outline

- General principles
- Architectural specification
- Software environment
- Current status

GENERAL PRINCIPLES

General Principles

- Layering
- Modularity
- Iteration
- Object orientation

Layering

- Layering separates the software components into a hierarchy with the application at the top, the domain in the middle, and the technology at the bottom. Each layer uses the layer below it, but not vice versa.
- Layering encourages the reuse of software components in different parts of the application.
- Layering provides an integrated framework for the software development.

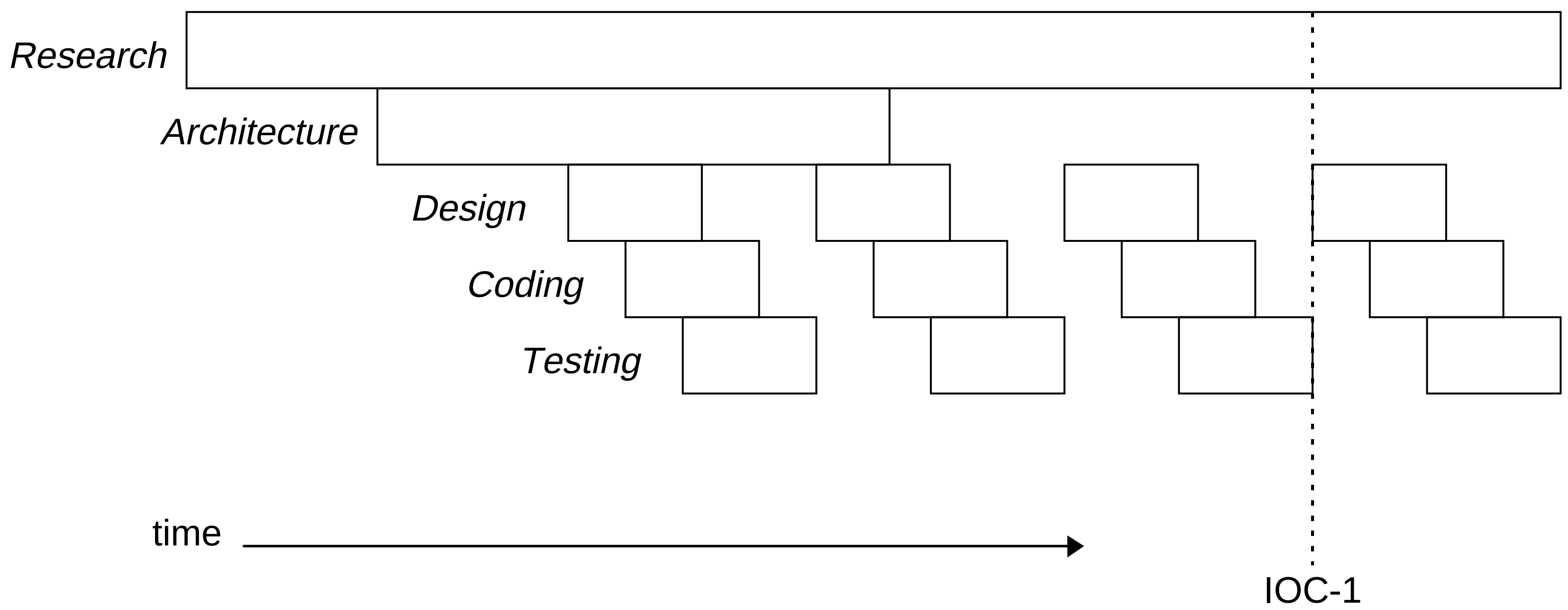
Modularity

- Each software component/module has responsibilities and provides services. The actual implementation of the module is separate from its public interface.

- Modularity reduces the coupling between software components that can make maintenance, reuse, portability, and extension difficult.

Iteration

- The iterative development process reduces risk.
- Each iteration refines the system and results in an executable release.



Object Orientation*

- Key features of object technology:
  - *Abstraction/Encapsulation* allows building models which map to the real world.
  - *Inheritance* enables code reuse.
  - *Polymorphism* reduces maintenance and increases extensibility.

- Critical issues addressed by object technology:
  - Scheduling (meeting delivery dates)
  - Complexity (modeling complex applications)
  - Size (managing interdependencies in large systems)
  - Compatibility (making different chunks of code interoperate)

---

* *After J. Berry, 1994.*

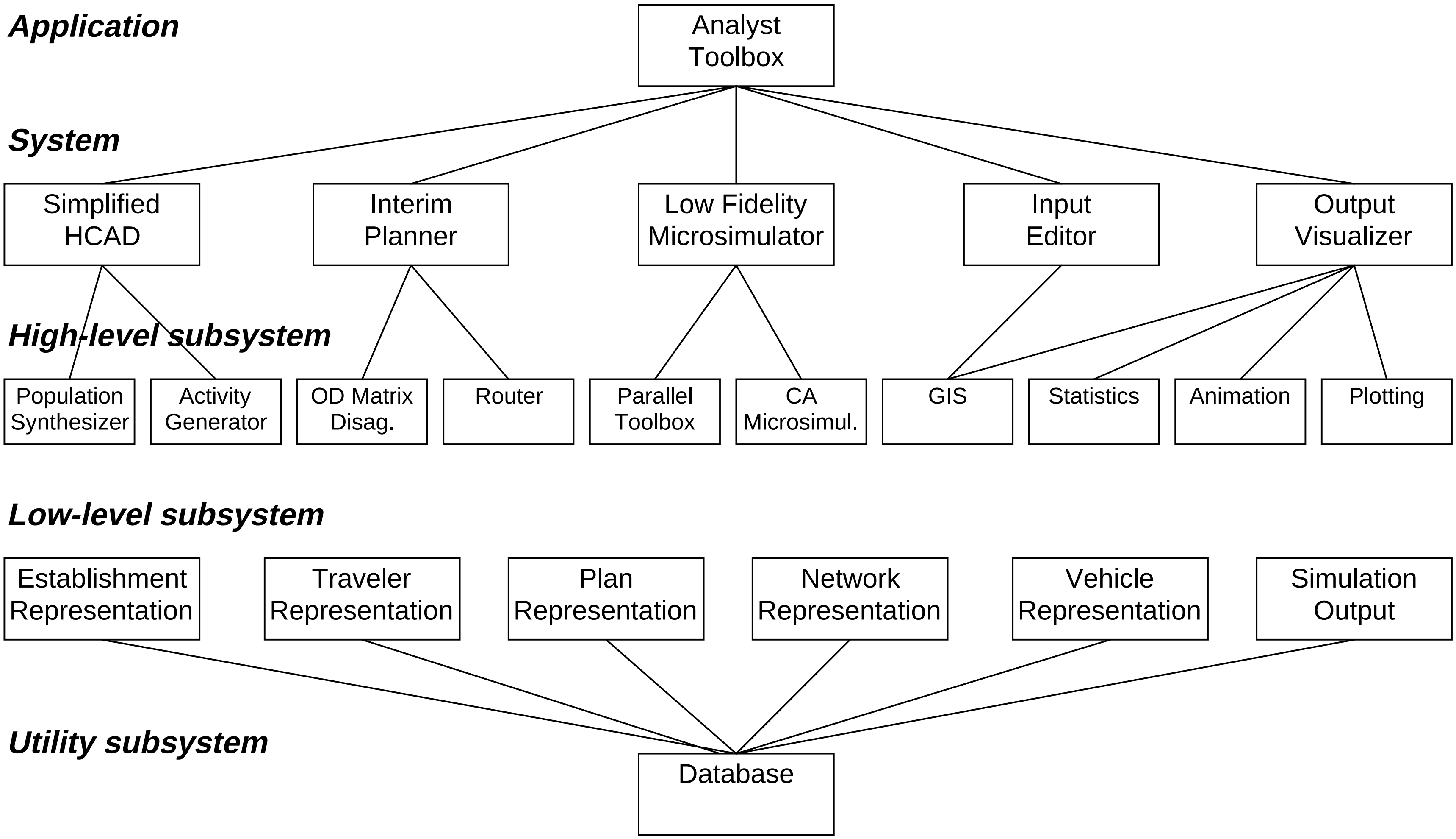
- This results in reuse of code, reduced code size, increased productivity, and lower defect rate.

ARCHITECTURAL SPECIFICATION

Architectural Specification

- Application layer
- Systems layer
- High-level subsystems layer
- Low-level subsystems layer
- Utility subsystems layer

TRANSIMS Architecture



Application Layer

- The Analyst Toolbox provides a centralized interface to TRANSIMS.

Systems Layer

- The TRANSIMS systems centralize access to the major functional components of TRANSIMS.
- For IOC-1, there are the following systems:
  - Input Editor
  - Simplified Household and Commercial Disaggregator
  - Interim Planner
  - Low Fidelity Microsimulator
  - Output Visualizer
- Additional systems may be added for future IOCs or from sources external to LANL. For example:
  - Air Quality Estimator
  - High Fidelity Microsimulator
  - Land Use Estimator

High-Level Subsystems Layer

- The high-level subsystems each provide services to one or more TRANSIMS systems.
- For IOC-1 there are the following high-level subsystems:
  - Population Synthesizer
  - Activity Generator
  - OD Matrix/Route Disaggregator
  - Router
  - Parallel Toolbox
  - CA Microsimulation
  - GIS
  - Statistics
  - Animation
  - Plotting

High-Level Subsystems Layer (continued)

- These subsystems enhance the reusability and flexibility of the software.

  Example: The router subsystem can be used for planning by the Interim Planner and for re*planning by the Microsimulator.*

Low-Level Subsystems Layer

- The low-level subsystems provide basic services (mostly data and operations on data) to the high-level subsystems and to the systems.
- They supply a common representation of objects that are used throughout TRANSIMS.
- These subsystems never interact directly with the user.
- For IOC-1 there are the following low-level subsystems:
  - Establishment Representation
  - Traveler Representation
  - Plan Representation
  - Network Representation
  - Vehicle Representation
  - Simulation Output

Utility Subsystems Layer

- The utility subsystems provide basic domain-independent services to the higher-level components of TRANSIMS.

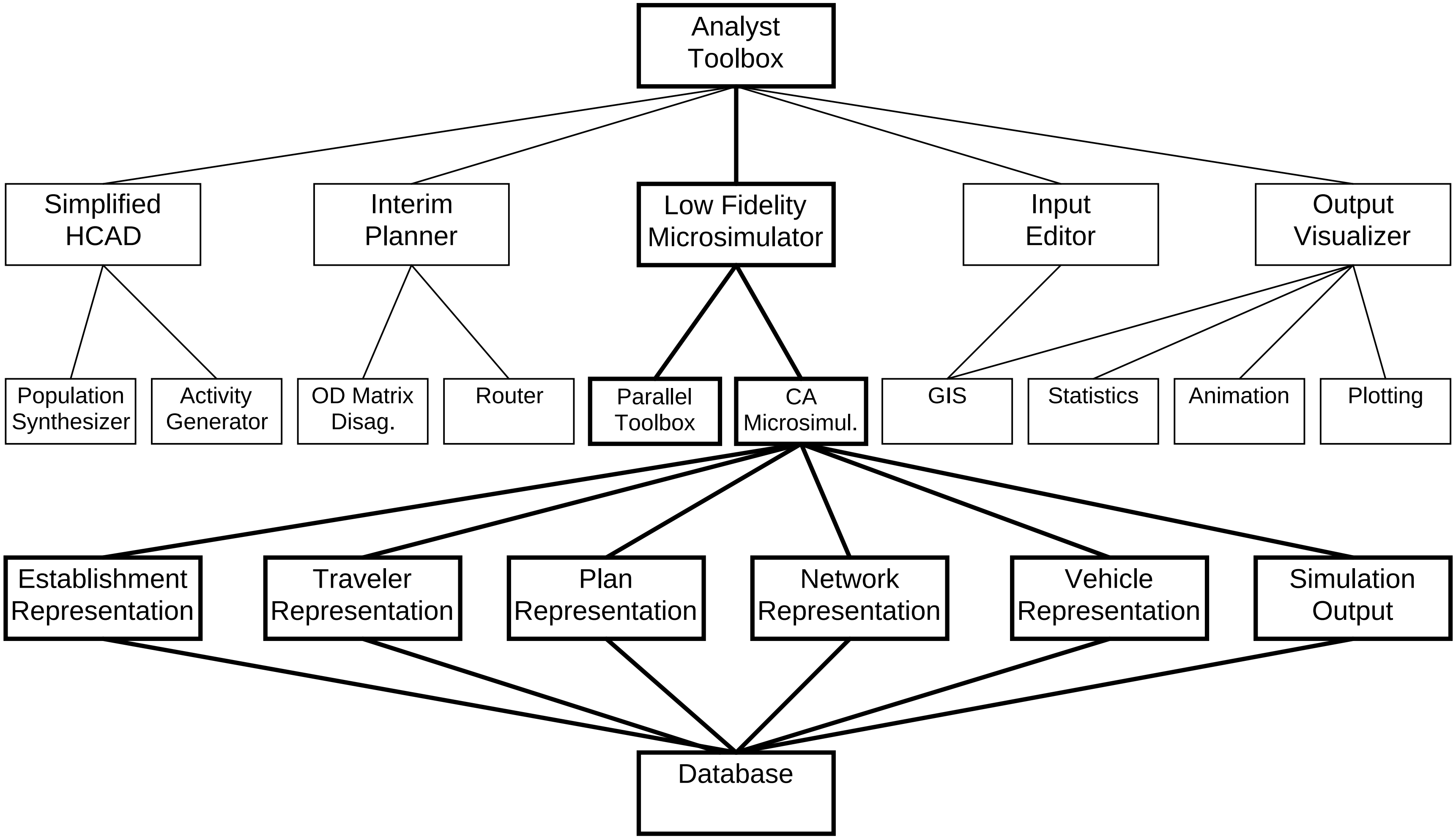
- These are used to isolate the rest of TRANSIMS from dependence on the specifics of the operating system, file system, communications network, etc.

- The components of this layer include:
  - Database
  - Interprocess Communication
  - Containers

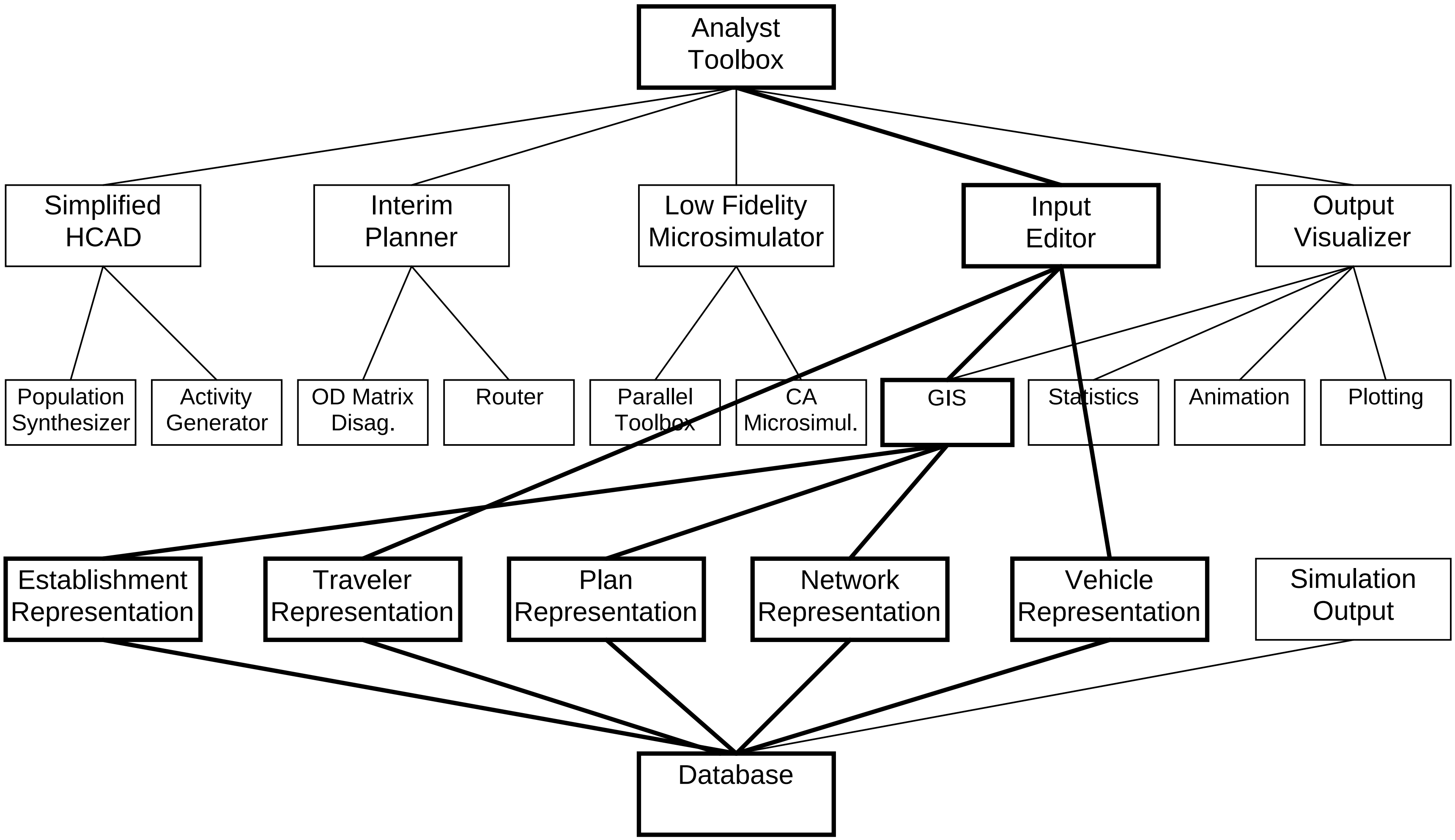
How the Architecture Works

- The application (i.e., the Analyst Toolbox) provides the user with access to the systems.
- Each system in TRANSIMS uses the subsystems as building blocks to accomplish its tasks.
- Different systems use different combinations of subsystems.

Software Components Used in Microsimulator



Software Components Used in Input Editor



Input Editor System

- The input editor provides a means for editing the database and setting up scenarios for simulation. All scenario data is editable. There are functions for:
  - importing new road networks
  - altering existing networks
  - merging networks
  - extracting networks
  - editing traveler characteristics
  - editing establishment characteristics
  - editing vehicle characteristics

- The editor is integrated into the rest of the GIS software, supporting visual/graphical editing of geographic objects, table editing of non-geographic objects, and editing via ad-hoc queries. Version is also supported.

Simplified Household & Commercial Activity Disaggregator System

- The simplified HCAD system provides the ability to construct travel activities for study areas.
- This system has the capability to select and import any of the available synthetic populations for a study region and to generate travel activity from the chosen synthetic population.
- The focus in IOC-1 is on households, as opposed to commercial activity.

Interim Planner System

- The Interim Planner provides the microsimulator with household or individual travel demand in the form of trip plans.
- For IOC-1, the major effort is to develop the data structure which integrates the “flow” of data from the synthetic population and its activity demand, through the planner’s trip planning process, and on to the microsimulation. The planner’s process, or mode/route/activity assignment algorithms, is simplified to accommodate available data and microsimulation development time constraints.
- Output data are the individual trip plans required by the microsimulation. This data includes trips and trip chains consisting of origins, activity destinations, routes and times associated with activity performance and route movement.

Low Fidelity Microsimulator System

- The low fidelity microsimulator is a regional-scale, low-fidelity traffic microsimulation based on cellular automata and implemented on a distributed computer network.

- The user interface to the low fidelity microsimulator provides a means for specifying CPNs, load balancing parameters, diagnostic outputs, and simulation outputs, as well as data sources to be used as inputs.

- This system is the primary technical focus of IOC-1.

Output Visualizer System

- The output visualization system provides an integrated interface to the various tools available for viewing and analyzing aspects of simulation and plan data. There are five types of output available:
  - animation
  - statistical analysis
  - geographic analysis
  - plots
  - data export

- The user can select a geographic region of interest (and time interval) and filter the available data before visualizing it. The output visualization system retrieves the specified data from the data sources and then sends the data to the appropriate visualization tool.

SOFTWARE ENVIRONMENT

Software Environment

- Platforms
- User Software
- Development Software
- Overview of Executables

Platforms

- The initial development of TRANSIMS is taking place on the Sun hardware/software platform.
- The development work is portable to other Unix platforms, but not necessary immediately, insofar as C++ compiler technology currently varies widely from platform to platform.
- Windows NT is supported for the user interface, but not for the planner and microsimulation.
- Windows, DOS, and MacIntosh are not directly supported, although they may be used as X clients for the user interface.

User Software

- ArcView is the primary graphical user interface. (Approximate cost $1500 per UNIX floating license or $900 per PC single-user license.)
- ArcInfo is used for the processing and analysis of geographic data. (Approximate cost $7100 for UNIX server with floating license.)
- The Oracle relational database is used to manage the majority of the data in the database subsystem. (Approximate cost $7000 for UNIX server with floating license and $200 for each client license.) Note that Sybase, Informix, and Ingress can also be used in place of Oracle.
- S+ is used for statistical analysis. (Approximate cost $2000 per floating UNIX license.)

User Software (continued)

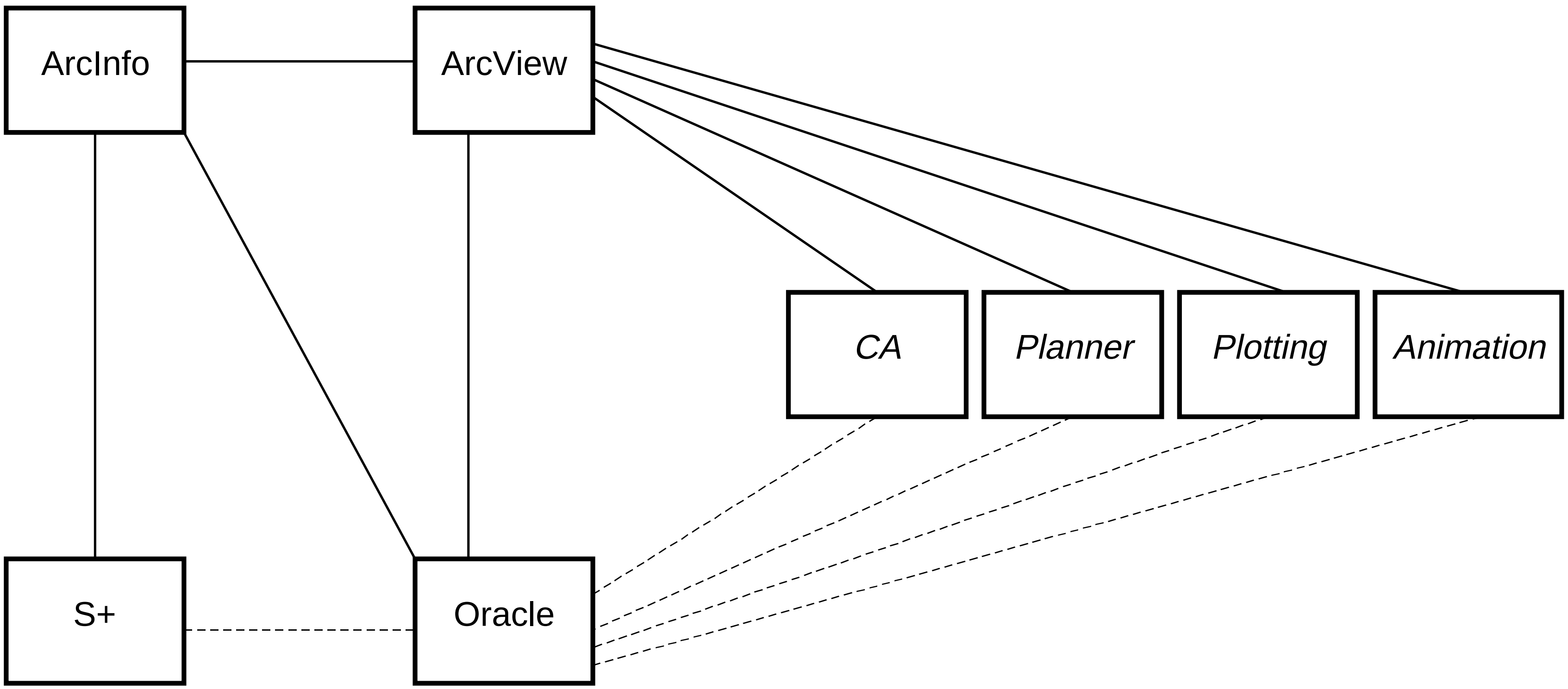
- Reasons for using commercial products:
  - Allows us to focus on the software development issues unique to TRANSIMS (the CA, Planner, etc.).
  - Speeds development time.
  - Enhances functionality
  - Many potential TRANSIMS users are already familiar with the commercial products we have chosen.
  - Extensive user support and training are available for commercial products.

- All of these products integrate with each other seamlessly for intercommunication and data use/transfer.
  - ArcView with ArcInfo
  - ArcView and ArcInfo with Oracle
  - ArcInfo with S+

Development Software

- All of the development software is portable to other UNIX and many non-UNIX platforms.
- The ANSI standard C library and the ANSI draft standard C++ library are used. POSIX-compliant operating system calls are also used.
- The Booch Components are used for container classes and exception handling.
- PVM is used for distributed application development.
- RPC is used to support interprocess communication.
- The Rogue Wave DBtools.h++ library is used for database access.

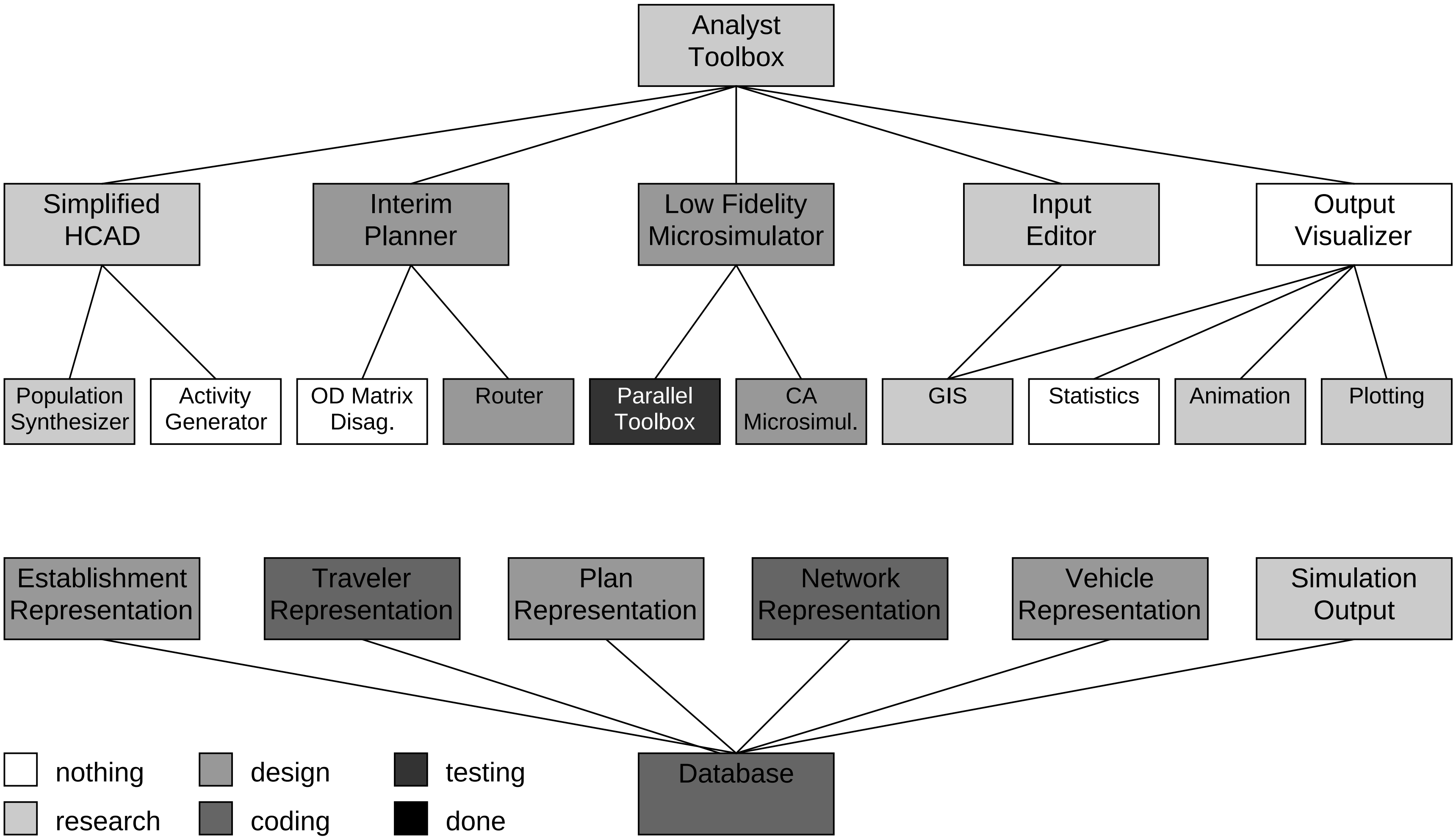
Software Executables Overview



In the above diagram, the solid lines represent control links and the dashed lines represent data links. The names in regular type are commercial products that are being customized; the names in italic type are executables that are being written in C++ by the software development team.

CURRENT STATUS

Current Status



Conclusions

- Architecture is very important in large software development efforts.
- TRANSIMS has a layered and modular design so that it is flexible, expandable, portable, and maintainable throughout its lifetime.
- The implementation of the TRANSIMS software modules is proceeding.

Network Representation Subsystem

The transportation network representation includes detailed information about roads, intersections, signals, sensors, transit systems, rail, bikeways and walkways. Network topology is represented along with attributes that describe the nodes and links in the network. Multiple views of the network representation are required, in some degree, by the Planner, Microsimulation, and Output Visualizer.

Link attributes for the road network include such characteristics as link type, length, directionality, speed limit, number of lanes, special lane designators, grade, toll, passing allowed, visibility range, street name, traffic capacity, etc. Node attributes may include node type, associated intersection (if applicable), etc. Provision will be made to represent the vehicular traffic network at a resolution at least as detailed as that of TRAF and TRANSYT-7F.

Intersections may be represented at multiple levels of fidelity. The Planner requires less fidelity than the Low Fidelity Microsimulator which uses a medium fidelity representation incorporating multiple queues. This intersection representation will include algorithms for handling the queues. Signalized and unsignalized intersections will be represented, and the representation must include lane usage and allowable turning movements. Both timed and actuated signals will be included along with sensors for actuated signals. The signal representation will include algorithms for cycling through signal phases.

Transit networks, associated transit schedules, and intermodal transfer facilities will not be represented in IOC-1, but provision will be made to include them in a later IOC.

The user interface to the network representation will be provided through the Input Editor. The network subsystem will support the importation of external data in formats such as Arc/Info,
TRANSYT-7F, and the TRAF family.

Establishment Representation Subsystem

The definition of an establishment includes households, group quarters, and businesses. Each establishment will posses a unique identifier and the socioeconomic attributes required by the activity generator subsystem.

IOC-1 will not focus on commercial activity or the movement of freight. Provisions will be made for the needs of the more sophisticated planner and disaggregator to be developed in a later IOC.

Traveler Representation Subsystem

The traveler representation includes the demographic, socioeconomic, and geographic attributes needed for identifying travelers and for planning trips. It also includes the driver representation and driver model needed for simulating driver behavior. The external data resources will not be available for the simplified planner. Thus the traveler representation for IOC-1 will be fairly simple. Nevertheless, it should implicitly specify unique travelers and make it possible to associate them with vehicle identifiers in the plan representation. The framework developed will be consistent with the future enhancements to be made in the planner and disaggregator.

The driver representation describes the attributes of the drivers used in the models. Driver's decision making processes may not be modeled at the same level of fidelity in every simulation, so a flexible driver representation that supports a particular decision logic without requiring unnecessary attributes is needed. Driver attributes will be specific parameters that are required by the decision logic algorithms rather than abstract behavioral attributes such as aggressiveness.

Potential attributes include driver age, sex, socioeconomic attributes, desired speed, following distance, and acceptable gaps for left and right turns, crossing intersections, and changing lanes. Algorithms that define the driver's decision logic are also part of the driver representation. Traveler attributes required by the planner include household id, mandatory activities, age, sex, socioeconomic class, trip goal weights, and mode/route preference distributions. Driver attributes included in the current CA Microsimulation include desired speed, and a variable that causes drivers to vary from their desired speed some of the time.

The user interface to the driver representation will support user specification of values for the attributes required in the study as well as the ability to read existing descriptions of driver properties.

Vehicle Representation Subsystem

The vehicle representation describes the attributes of the vehicles used in the models. Not all potential attributes are needed for every type of study, e.g. air quality studies require dynamic information about engine properties, while other studies may not. The vehicle representation should be flexible enough to support a variety of studies but not require unnecessary attributes for a particular study.

Potential attributes include such properties as vehicle type, maximum speed, maximum acceleration, maximum deceleration, stopping distance, length, width, weight, age, fuel type, and engine properties. Dynamic attributes such as position along road segment, lane, velocity, acceleration, and engine temperature may also be required. Algorithms that define the motion of vehicles are also part of the vehicle representation.

The version of the Planner to be used in IOC-1 requires only vehicle type. The current CA Microsimulation requires only maximum speed, but length will also be required if multiple vehicle types are supported. Acceleration and deceleration parameters may also be desirable to smooth speed fluctuations.

The user interface to the vehicle representation is provided through the Input Editor and will support user specification of values for the attributes required in the study as well as the ability to read existing descriptions of vehicle properties. When multiple types of vehicles are modeled, fleet mix will also be under user control.

Plan Representation Subsystem

The plan representation will provide a view of trips and trip chains in the form of routes, origins, and destinations during specific time periods.

A trip chain is a sequence of trips. A trip must provide a:

- trip purpose
- unique identifier for the traveler
- unique identifier for a vehicle
- starting node in the network
- desired departure time from that starting node
- destination node in the network
- desired arrival time at that destination node

In the microsimulation, detailed routes will specify which streets simulated vehicles will follow and when they should be at various points along the way to satisfying an associated trip goal.

The detailed route design will impact the complexity of both the microsimulation and the visualization systems. The actual format might not include the ID of every link and node along the way. Perhaps a detailed route should be included as part of the trip specification.

Population Synthesizer Subsystem

Using 1990 census data a baseline synthetic populations of households will be generated which statistically mimic those sampled in the 1990 census. These populations will be produced on a census tract or block group basis. Each household and person in the synthetic population will be associated the entire suite of socioeconomic characteristics available from the census. For future applications these households will be aged to the desired date using projected land use and demographic trends in the study region. The populations will not be aged for IOC-1.

Input data for the generation of the baseline population will include the Census Bureau Standard Tape File 3 and the Public Use Microdata Sample. Standard statistical techniques will be used to generate the synthetic populations.

For IOC-1 baseline synthetic populations will be generated off-line. Multiple populations will be produced, but the software for doing so is not part of IOC-1. The households of these populations will be placed at random at locations in the census tract or block group.

Activity Generator Subsystem

Travel activity will be predicted for each household and household member using national trends and local activity surveys. These desired travel activities will be passed to the planner for routing and scheduling. Activities will be assigned to either the household or the individual. For example work activities would be assigned to the individual while a household activity could be a shopping trip. Additionally, activities will be either mandatory or discretionary.

The structure of IOC-1 will allow for future activity based travel. However, for the applications considered in IOC-1, travel activity will be replaced by trips which are generated from OD matrices. All trips will be assigned to individuals (no household activity analysis is planned for IOC-1) and all trips will be treated as mandatory. Trips from an OD matrix will be matched with the demographics for the households. These will be randomly distributed to individuals from demographically matched households within the census tracts or block groups which make up the Dallas Travel Survey Zones.

OD Matrix/Route Disaggregator Subsystem

For IOC-1, actual travel data is available only in the form of OD matrices and, possibly, traffic and turn counts. To produce trips, a utility preprocessor will be needed to disaggregate OD matrix zonal based traffic flow down to individual travelers, specific routes and activity addresses. Methods are currently being examined which will disaggregate the zonal flows along the boundaries of the detailed area of interest (e.g., along the LBJ corridor). Once the individual travelers are placed on routes which enter the area of interest and assigned "start" times, the Simplified Planner will then perform the normal route assignment to generate detailed individual trip plans. The travelers will move to final destinations within the area of interest or through and out of the area of interest based upon their aggregate OD matrix assignments. If traffic or turn counts are available for the area of interest, they will be used to calibrate the planner’s preference distributions.

Router Subsystem

Given household or individual activity demand, individual travel behavior, and individual travelers along the boundaries of the detailed area of interest, this subsystem will provide feasible or "optimally" infeasible sets of trip plans. A feasible trip plan is one in which all individual goals have been satisfied; an "optimally" infeasible plan is one which has not satisfied all of the individual’s goals but which has minimized the non-zero goal deviations.

The router will be structurally consistent with the mode, route and activity assignment enhancements planned for later IOCs.

Parallel Toolbox Subsystem

The parallel toolbox subsystem is those parts of the Low Fidelity Microsimulator that deal with running on a parallel distributed computer. The parallel toolbox provides a master/slave parallel computing model implemented on top of the PVM toolkit for heterogeneous network computing. PVM provides the message-passing substrate that allows tasks on different machines (CPNs) to communicate. Using the parallel toolbox, dynamic load balancing of the road network and associated vehicle objects is provided and is based on usage statistics gathered while the simulation runs. CPNs may be added or deleted during a running simulation. Diagnostic outputs will be provided as will some form of fault tolerance. Simulation outputs will be produced in parallel.

CA Microsimulation Subsystem

The CA microsimulation subsystem is those parts of the Low Fidelity Microsimulator that deal with doing traffic simulation using cellular automata. The network representation in this approach is grid based, and a mapping from the general network representation subsystem to grids will be provided. Similarly, mappings from the vehicle and traveler representation subsystems into vehicle/driver combinations that are suitable for the CA approach to vehicle motion will be made. Travelers will utilize plans from the plan representation subsystem.

Simulation Output Subsystem

The simulation output subsystem will gather the data generated by simulations and provide access to it for other subsystems as soon as the data is received. The simulation output will be configurable and several predefined configurations will be provided:

trajectory information (time, segment, lane, position along segment, velocity, acceleration)

control systems (signals, sensors, HOV lanes)

vehicle state information (brakes on, lights, signaling)

measures of effectiveness (VHT, VMT, average speed, average density, headway)

animation output (trajectory and control systems)

engine performance (idle time, start time, stop time, temperature, fuel consumption)

emissions (CO, NOx, O3, particulates, aerosols) traveler characteristics (vehicle occupancy, demographics, trip purpose, plan fulfillment)

additional outputs available from TRAF and TRANSYT-7F products

This subsystem will utilize the database subsystem to support metadata, data distribution, data export, and archiving. Special provision will be made for dealing with the large amount of data generated by simulations. It will also be possible to perform compression (lossy or lossless) on the data to reduce the storage required for it. (Only limited capabilities will be developed in this area for IOC-1.)

In IOC-1, no automated feedback mechanism will be provided for using simulation output to alter simulation or planner input for subsequent simulations. This can be accomplished, manually, through a (possibly complicated) series of ad-hoc queries involving the output and input data sets.

GIS Subsystem

The GIS subsystem provides high-level support for the following functions:

editing network data

viewing all types of geographic data and non-geographic data that can be linked to geographic data

thematic display and analysis of plans and simulation output

aggregation/disaggregation of geographic data

preprocessing/formatting of geographic data from external sources (i.e., MPO data)

The GIS subsystem will support the import/export and editing of data available from other data-related subsystems (database, network, traveler, vehicle, plan, output) as well as support the export of data to visualization and analysis tools such as the plotting, statistics, and animation subsystems.

Statistical Analysis Subsystem

This subsystem will support the following general types of statistical analyses:

- confidence intervals
- analysis of variance
- hypothesis testing

All of the data available from other data-related subsystems (database, network, traveler, vehicle, plan, output, GIS) will be importable for analyses. Predefined and user-definable analyses will be available for computing various measures of effectiveness; it will also be possible to save an analysis configuration for later recall and re-running. The results of analyses will be exportable to the plotting subsystem.

Animation Subsystem

This subsystem will provide animated display of vehicle movement and traffic controls (e.g., signals) in real time and in accelerated time. Other map features such as buildings, bodies of water, topography will not be displayable for IOC-1. The vehicles and network features will be selectable with the mouse to obtain detailed attribute data. It will be possible to color-code vehicles based on their attributes. Plans can also be animated.

Plotting Subsystem

This subsystem will support the following general types of plots:

- scatter plot
- histogram
- lines
- areas

Three-dimensional and color plotting will be supported, as will be the grouping of attributes. All of the data available from other data-related subsystems (database, network, traveler, vehicle, plan, output, GIS, statistics) will be importable for plotting. Predefined and user-definable plots (whose configuration can be saved and loaded) will be available. It will also be possible to customize axes, legends, markers, and annotation.

The specific types of plots will include the following:

waterfall plots
fundamental diagrams (all styles)

Database Subsystem

The database subsystem provides low-level services for accessing and modifying data. It forms a layer separating the other subsystems from the actual data files-the other subsystems will not have access to the data files at the physical level.

Each data source will be indexed by a unique (primary) index. Additional (secondary) indexes will be allowed. The interaction between other subsystems and the data will be mediated by the public interfaces of the classes in the database subsystem. Range lookup of key values and ad-hoc SQL queries will be supported.

Procedures (e.g., C++ templates, preprocessor macros, or custom preprocessor) will be available to coordinate the maintenance of the database schema and class definitions.

The database subsystem will maintain metadata specifying the following for all of the data sources:

existence
versions
network location
attributes (data dictionary)

The distribution of a data source over the network will not be supported for IOC-1, although this capability will be added in a later IOC. Migration of data will also be supported, but the database subsystem will not be required to automatically load-balance the data among the network nodes. An archiving facility will also be available.

There will also be a mechanism for extracting data specified at run-time from a data source and exporting it to a formatted binary or ascii file.